Reliability of the forecasting of the monthly demand of electric energy with neural networks

M. A. Jaramillo¹, D. Carmona², E. González² and J. A. Álvarez²

 ¹ Department of Electronics and Electrical Engineering E.I.I., University of Extremadura Avda. de Elvas s/n, 06071 Badajoz (Spain)
phone:+34 924 289600, fax:+34 924 289601, e-mail: miguel@unex.es

² Department of Chemical and Energy Engineering

E.I.I., University of Extremadura Avda. de Elvas s/n, 06071 Badajoz (Spain)

phone:+34 924 289600, fax:+34 924 289601, e-mail: dcarmona@unex.es, evagzlez@unex.es, jalvarez@unex.es

Abstract. Electric energy demand forecasting represents a fundamental tool to plan the activities of the companies that generate and distribute it. So a good prediction of its demand will provide an invaluable information to plan the production and purchase policies of these companies. This demand may be seen as a temporal series when these data are conveniently arranged. In this way the prediction of a future value may be performed studying the past ones. Neural networks have proved to be a very powerful tool to do this. They are mathematical structures that mimic that of the nervous system of living beings and are used extensively for system identification and prediction of their future evolution. In this work a neural network is presented to predict the evolution of the monthly demand of electric consumption. A Feedforward Multilayer Perceptron (MLP) with three hidden layers has been used as neural model with Backpropagation as learning strategy. The consumption data have been normalized to avoid their rising trend. Several procedures have been tested in order to find out those performing the best. Errors smaller than 5% have been obtained in most of the predictions.

Key words

Demand forecasting, neural networks, data normalization, electric consumption, time series prediction

1. Introduction

Electric energy demand forecasting is a fundamental tool for production and distribution companies because it provides them a prediction of the market needs of electric energy, so that they will be able to fit the electric energy production to the society demand. Two kinds of forecasting may be performed: a short term one that deals with prediction of hourly or daily consumption and a long term one that works with monthly data.

Short term prediction provides information about specific demands for a few hours ahead, which allows companies to adapt electric energy production to the society needs. Long term prediction provides both producers and distributors with a forecast of the evolution of the demand, which allows the definition of strategies to augment the capability of the distribution network, the construction of new production plants or the development of policies to obtain new clients.

As electric consumption data evolve along time they may be assumed to form a time series, where the prediction is done considering past values to forecast future ones. Nevertheless as electric energy consumption is highly influenced by factors different from past consumption, prediction needs to take into account, along with those past consumption values, other information as temperature, humidity, hour of the day or day of the week, because all of them have a remarkable influence on demand. All this information is essential when a short term prediction is to be done because the electric energy consumption depends on all those factors. In this way a great amount of data is needed to perform the prediction and complex models must be used to obtain good results.

On the other hand, as long term forecasting deals with monthly data, the influence of the aforementioned factors is diluted in an overall value. These values represent the total monthly electric consumption where specific peaks or valleys are diluted in the overall information considered, so their effect in the data is small. In this way any kind of isolated distortion in the demand will have little influence in its overall monthly value. Nevertheless, although their effect is not very important in the evolution of the time series, they generate fluctuations that are difficult to take into account because their causes are not provided. In any case, long term demand forecasting needs less accurate predictions than the short term one because it influences global decisions regarding overall productions or purchases where small fluctuations have little influence. So a less robust prediction may be done avoiding the use of the sophisticated tools that are needed to deal with short term predictions. In the same way, as the influence of weather conditions or the kind of the day (holiday or weekday) is diluted in the monthly

overall consumption, the forecasting model may use only past elements of the time series to obtain a prediction of the electric demand.

As it has been pointed out previously electric demand forecasting may be studied as a time series prediction problem, so tools which provide good results in solving this problem may be used to perform electric demand forecasting. Among them neural networks have shown to be an important tool because of their flexibility and easy configuration for solving the time series prediction problem, a fact that is hardly surprising if it is taken into account that recurrent neural networks may be considered as a special case of nonlinear autoregressive moving average models (NARMA), a very powerful tool for predicting the time series evolution [1]. So neural networks have become popular tools for the electric demand forecasting [2], [3], [4]. These works are mainly devoted to short term prediction, using the neural network capability to deal with different kinds of data as the variables to be processed [4].

Nevertheless in spite of its potential interest for electric energy companies, long term forecasting has received little attention from researchers in contrast with the higher interest that short term one has had [5]. In this work a monthly demand prediction is carried out by a neural network to study the performance of this kind of forecasting. The selected neural model is the Multilayer Feedforward Perceptron, one of the most popular neural structures.

Before programming a Neural Network it is necessary to study the structure of the data to be processed in order to fit them and the network inputs and outputs. In this way it may be seen (Figure 1) that time series of monthly demand have a rising tendency due to the influence of the economic and technological evolution that generates a growing demand of electric energy. This trend has a very important influence in the evolution of the time series, and its effect must be considered in the definition of the neural network.

Unfortunately neural networks have prediction problems when dealing with time series that present a tendency of this kind because of the use of nonlinear saturating functions as neural outputs that impose boundaries to the network outputs. So the model will not be able to predict values of the time series falling out of that boundaries. Therefore it is necessary to define a normalization process which transforms the original time series into another where the rising trend does not appear. The way the normalization is carried out represents a very important step in the definition of the neural network structure because it determines the amount of information that those data provide the network to perform a prediction. So a good selection of this procedure is fundamental for the network to provide the best performance. Unfortunately there is no way to decide which is the best normalization procedure for each problem and a trial and error strategy is to be used to find out those performing the best. So in this work some normalization processes have been tested and applied to the same network in order to determine the performance of each one.

The Spanish monthly consumption from January 1975 to December 2002 (a total of 336 values) has been used to validate the proposed model. All these data has been divided into two blocks: one for training (from January 1975 to December 1996, 264 months) and the other for validation (the remaining information, 72 months).

The outline of this work is as follows: Section 2 describes the structure of the neural network used to perform the forecasting. In Section 3 several normalization processes are described, while the results obtained with each of them are studied in Section 4. Finally, in Section 5 some conclusions are presented.

2. Neural Network structure for time series prediction

A. Structure of the Multilayer Perceptron

As it has been previously stated the neural network structure used in this work is the well-known Multilayer Perceptron. It has been widely used in time series prediction because of its ability to identify the time evolution of a dynamic system. Actually it may be assumed as a universal approximator [6].

The network is formed by an input layer whose elements are the data to be processed, an output layer that provides the output data of the network and one or several hidden layers that process the incoming information to obtain the network response at the output layer. As it can be seen the input layer is not actually a layer but rather the input data to the first hidden one. Every layer may be formed by a variable number of units named neurons. Each neuron computes the weighted sum of all its inputs and a bias constant. The result is processed by an activation function that provides the neuron output:

$$x_{j}^{i} = \sum_{k} w_{jk}^{i} y_{k}^{i-1} + \theta_{j}^{i}$$
(1.a)

$$y_j^i = \sigma\left(x_j^i\right) \tag{1.b}$$

In this expression x_j^i represents the activity of neuron jin layer *i*, w_{jk}^i the strength of the connections between this neuron and all those that are in the previous layer, y_i^i the neuron output while θ_i^i is a bias constant.

The output function $\sigma(x_j^i)$ represents an important element of the neural network paradigm, because it supplies a nonlinear element to the model that allows these structures to identify the nonlinear behavior inherent to complex dynamics. The most frequently used as output function is the so called sigmoid for its "Sshape" form. It may be the arctangent or hyperbolic tangent when it is needed an output included in the [-1,1] interval. When the desired interval is [0,1] the so-called logistic function is used:

$$y_{j}^{i} = \sigma \left(x_{j}^{i} \right) = \frac{1}{1 + \exp(-Ax_{j}^{i})}$$
 (2)

Nevertheless very simple algebraic manipulations allow the use of any of the two sigmoids with both output intervals. The use of other kinds of output functions like linear, piecewise linear or step is also usual.

A combination of both linear and nonlinear functions is usually found in the literature, where a very common configuration is the use of nonlinear functions for the hidden layers and linear functions for the output one. This configuration is based on the fact that the information processing is performed by the hidden layers while the output one usually provides only an adaptation of the neural network response to the desired size (number of outputs) of that response, moreover, the use of a nonlinear saturating function in the output layer will eliminate the possibility of signals with higher than saturation values at the output, a fact that will diminish the precision of the predictions performed by the network. This is the configuration selected for the model proposed in this work.

Another important issue in the definition of the network structure is the connection scheme of every neuron. A fully connected feedforward network has been selected, where each neuron is connected to every output of the previous layer and no connection is allowed between neurons in the same layer. It has proved to have very good capabilities in the approximation of any nonlinear function, and also in time series prediction [6]. The inclusion of feedback will need the demonstration of the network stability and will not endow the model with further capabilities, therefore it is usually not included in neural network models used to identify time series.

B. Learning process

The learning capability of neural networks is provided by the adaptation of the input weights of every neuron. This process is performed by presenting an input pattern and the desired output to the network and then modifying every weight until an error function reaches a minimum or falls below a fixed value. This procedure is repeated for each pattern to be learned. The way this adaptation process is carried out defines the learning strategy of the neural network. The well known Backpropagation algorithm [7] has been selected to do it. It uses the mean squared error as error function. So, once an input pattern is presented to the network it provides an output whose value is compared with the desired one and an error function is obtained. With this function it is possible to calculate an expression that relates the gradient of the overall error function to every weight. Several algorithms may be used to perform this calculation. The selected one is the so-called Levenberg-Marquardt (a combination of the gradient descent and Newton methods for solving optimization problems) [7] which provides a very accurate weight adaptation with a moderate time consumption. Its only drawback is the need of a very large memory in the computer where the process will be simulated, an easy to accomplish requirement in modern computers.

C. Network Structure

The number of neurons in the hidden layer is an essential issue in the network design strategy. So if a network is provided with too few neurons it will not be able to reproduce the system dynamics accurately and therefore it could not provide a reliable forecasting. On the other hand, too many hidden neurons will define a network that, in the best case, will provide an appropriate behavior but with an excessive computing time and a high memory need while, in the worst case, will only learn the presented patterns and will not be able to generalize the acquired knowledge to predict non-learned patterns. Therefore the selection of the appropriate network size is not an easy task. Some algorithms have been developed to look for the best dimension of the network, the so-called pruning algorithms [8]. Their working strategy is very simple: it starts with the definition of a larger than necessary network and then proceeds to simulate its behavior to detect redundant neurons or links that will be next removed. The process is iteratively repeated while an error function is kept under a certain value. These algorithms usually consume too much time and not always provide the smallest network size

An easier approach may be done with a trial and error strategy, where several networks with different sizes are simulated and that with the best performance is selected. If a more precise response is needed variations about this size will be tested until the desired precision is obtained.

Along with the definition of the network size it is also important to determine the number of hidden layers it has. Sometimes the use of a multilayer network provides better results than using an only one, others an only layer will be enough. Here there is no algorithmic procedure to obtain the best solution and a trial and error strategy is to be used.

Several layer numbers and sizes have been tested to find out the best one [5]. Some one layer and multilayer structures have been tested with one, two and three hidden units. The most significant ones were (8) and (12) for a one layer network, (4,2), (6,2), (6,4), (8,4) and (12,6) for a two layers network and (4,2,4), (8,4,8) and (12,6,12) for a three layers structure. The network that presented the best results was (8,4,8). Thus, this structure has been used to test different kinds of normalization processes, as described bellow.

3. Input data structure

As it has been previously stated time series of electric energy demand present a rising tendency due to the influence of the economic and technological evolution on the electric market. In fact, as the wealth of most nations presents a rising tendency and the technological development provides the society with more and more devices that need electric energy to work, the electric demand suffers a constant rising tendency. This general rising trend clearly appears in the example used to test the presented model (Fig. 1). Imbedded in this general trend a fluctuation caused by the difference in demand that each month has appears. The electric demand in the same month of different years must have very similar variations from the general rising tendency because this demand is mainly controlled by climatological factors, which must be very similar every year. Modifications in those factors are usually embedded in a more general climatological behavior that will influence several months. In the same way specific political or economical factors different from those that govern the general rising tendency of the demand will influence several months. So variations in normal demand will affect several elements of the time series, providing a slight modification of the general rising trend. Those variations may be properly handled if enough past values are considered to perform a prediction. Then it may be assumed that the time series is formed with the combination of a general rising tendency and monthly variations that ought to be very similar for different years.



Fig. 1. Monthly Spanish electric consumption from January 1975 to December 2002.

That rising trend represents an important problem for neural networks that are to be used to perform the demand forecasting because the use of nonlinear saturating functions as neural outputs imposes boundaries to the network outputs that avoid the network providing an output higher that a certain upper limit. So the model will not be able to predict values of the time series falling out of that boundaries. The use of linear output functions in the output layer does not solve this problem because saturating functions in the inner layers will still impose an upper limit to the network output. Removing all the saturating functions from the whole network is not a solution because in this way a linear network will be defined and it will not be able to follow the highly nonlinear behavior of time series that represent the electric energy demand. Therefore it is necessary to define a procedure that eliminates that rising tendency from the time series but retains the monthly fluctuations. In this work this procedure is defined as a normalization process which transforms the original time series into another where the rising trend does not appear and only the monthly evolution is retained. Nevertheless there is no beforehand way to define an optimal procedure to perform that normalization process, therefore several strategies must be tested in order to find out the one performing the best.

As the normalization process try to extract the monthly deviation from the general rising tendency of the time series it will be necessary to obtain a kind of mean value that represents that rising trend and a deviation from it that will be assumed as that monthly deviation. Since the prediction problem deals with the forecast of a value from those preceding it the normalization process will use only past values to do it. So a set of values preceding that to be normalized will be considered to perform a certain mathematical operation to obtain a general trend linked to that element. This calculation will be applied to every element of the time series except those n at the beginning, where *n* is the number of elements taken into account to obtain the normalization value (this is so because it is not possible to consider *n* values to obtain the normalization constant when there are only n-kelements preceding the considered one).

Once every normalization constant has been obtained every consumption will be divided by its corresponding constant to obtain a deviation associated to every trend. A division has been preferred to a subtraction because with the last it is possible to obtain very high values along with others too close to zero and the normalization process will result little effective. On the other hand the division will provide a value that will never be too high (smaller values may be close to zero, although the highest ones will never be much higher than one). In this way, the values obtained with the division are well suited to be processed by a neural network whose outputs and weights are included in the interval [-1,1] ([0,1] for the output function when it is a logistic one).

In this way two new time series have been defined: one with the general trend associated to each consumption datum and another with its deviation from that value. This last one will provide the inputs to the network.

In order to fit the input data structure to the network layer structure, in most cases, the number of elements taken into account to perform the normalization process will be the same as the number of values in the input layer. Although it is not necessary to impose this condition to the definition of the input layer it looks meaningful in order to warrant a certain homogeneity between the normalized data and the set of them that are presented to the network every time a prediction is to be performed.

In this work the following normalization processes have been defined:

1) The data to be normalized and a set of those preceding it are added up to form the normalization

constant. Several options have been tested in this way:

- Six values are presented to the network. Every datum is divided by the sum of the preceding five along with itself. Here the seasonal influence in the monthly demand is taken into account.
- Twelve values are presented to the network. The normalization constant is obtained in a way similar to that in the previous option with the only difference that eleven past values are considered. Here a yearly data evolution is considered, because as the electric demand presents similar monthly patterns for different years it is reasonable to assume that a normalization that spans a year back will extract that monthly evolution from the overall yearly tendency.
- Thirteen values are presented to the network. Every datum is divided by the sum of the preceding twelve along with itself. This is similar to the previous option, but it includes the influence of the corresponding month of the previous year.
- 2) The datum to be normalized and several values of the same month of preceding years are added up to form the normalization constant. It allows knowing the tendency of each of the months of a year, in the context of the corresponding year.
- 3) The normalization constant is obtained as the average value of a group of the preceding data and that to be normalized. The same options considered in the first case have been tested.

As the network input data are normalized the corresponding prediction is also normalized, therefore a process opposite to that one is to be performed in order to obtain the corresponding forecasting of the actual demand. So the corresponding denormalization constant ought to be obtained in the same way to that used in the normalization process. Nevertheless it is not possible to use the actual value of the predicted demand to obtain its corresponding denormalization constant because it is unknown. But as the normalization constants represent the rising trend of the time series two adjoining elements will have very similar values and a very small error is made if one is used instead the other. So it will be possible to use the normalization constant of the preceding element as denormalization constant of the predicted one. That is to say, if element t+1 of the time series is to be obtained, it will be multiplied by the normalization constant of element t.

Finally, a corrective process has been applied to the resulting values because, although the neural network can learn the behavior of a time series, it cannot predict an unexpected change in the series tendency until enough values of the monthly demand have been presented to the network.

To compensate for this effect a new variable is defined. It represents the difference between the denormalized value and an expected value for it. This one cannot be the real datum, because it is not known when the prediction is performed, and an estimation is to be obtained for it: the same month of the preceding year multiplied by the unit variation between subsequent years. Equation (3) shows this variable:

$$C(i) = P(i) - R(i-12) \times \frac{R(i-1) - R(i-13)}{R(i-13)}$$
(3)

where C(i) is the corrective variable of the *i* datum; P(i) is the predicted value of the *i* datum; R(i) is the real value of the *i* datum.

As equation (3) shows, the expected value for the i datum is estimated from the real twelve-months-before value multiplied by the demand variation that has occurred through the previous year.

This correction is weighted by a factor 0.6, a value that has been obtained by a trial and error procedure which provides the best results among all those tested between 0 and 1. So a factor 0.6C(i) is added up to the denormalized datum to obtain the network prediction.

In this way the network provides a better output when the trend of the time series changes, while negligible variations appear with data with a uniform trend.

4. Computational results

As it has been previously stated the Spanish monthly consumption from January 1975 to December 2002 (a total of 336 values) has been used to validate the proposed model. All these data has been divided into two blocks: one for training (from January 1975 to December 1996, 264 months) and the other for validation (the remaining information, 72 months).

The learning algorithm had a maximum number of 5000 time steps to reach an adequate error (between 0.0001 and 0.000001), a preventive measure to avoid the algorithm falling into a too long learning process, although in most cases the desired error was reached in less than 1500 time steps, which proves the speed of the selected algorithm.

The best results have been obtained using the average value of six or twelve values as the normalization constant. Although the error during the learning process is limited to 0.0001, results show only seven errors over 5% when six values were used. Four errors higher than a 5% were obtained when twelve values were used, although they were greater. Fig. 2 and Fig. 3 show the normalized values and the resulting errors, respectively, for the 6-values normalization, whereas Fig. 4 and Fig. 5 present the corresponding normalized values and errors for the 12-values normalization.



Fig. 2. Normalized data using the average six values as normalization constant.



Fig. 3. Simulation errors using the average six values as normalization constant.



Fig. 4. Normalized data using the average twelve values as normalization constant.



Fig. 5. Simulation errors using the average twelve values as normalization constant.

The other normalization processes proposed in Section 3 were also simulated, but worse results were obtained. The results are summarized in Table I. The table presents a global error associated with each normalization process. It corresponds to the network training goal. They are different because the simulation process has shown that in some cases a too reduced error can produce the data memorizing, losing generalization ability. So a rising of its value was necessary in order to improve the forecasting precision. Those cases in which the normalized data are very small (those whose normalization constant is the sum of real values) can work with a smaller error.

NORMALIZATION CONSTANT	ERRORS GREATER THAN 5%	GLOBAL ERROR
Sum of 6 values	8	0.0000012
Sum of 12 values	11	0.000001
Sum of 13 values	10	0.000001
Average value of 6	7	0.0001
Average value of 12	4	0.0001
Average value of 13	9	0.0001

Table I. Results obtained in each case

The results obtained with Option 2 have not been included in Table I, due to the bad result they present. The causes of these bad results are the values taken into account for the denormalization process. These values are the same month of the past years, which are, in general, lower than those used for the normalization (the same month plus the same of the preceding years), due to the rising tendency of the series. The correction process previously described is not able to compensate for this effect.

5. Conclusion

A neural network has been presented to perform electric demand forecasting that takes into account only past monthly demand data to obtain a prediction of the following month. Some neural structures have been tested with a variable number of layers and neurons in each layer. The one performing the best was that with three hidden layers with a (8,4,8) distribution of neurons. As input data present a rising trend a normalization process was necessary in order to fit this data to the network input and to eliminate that rising trend from the time series. Several normalization processes have been proposed and then tested in order to find out those performing the best. These were those that divide every consumption by an average of past values of the time series. These results show the good performance of neural networks to forecast long term electric demand. It has been also highlighted that an appropriate representation of input data is essential in order to obtain the best performance of the neural model.

References

[1] J. T. Connor, R. D. Martin, L. E. Atlas. "Recurrent Neural Networks and Time Series Prediction". IEEE Trans. on Neural Networks, Vol. 5, No. 2, pp. 240-254, March 1994.

- [2] W. Charytoniuk, E. Don Box, W.J. Lee, M.-S. Chen, P. Kostas, P. Van Olinda. "Neural-Network-Based Demand Forecasting in a Deregulated Environment". IEEE Trans. on Industry Applications, Vol. 36, No. 3, pp. 893-898, May/June 2000.
- [3] F. Doveh, P. Fegin, D. Greig, L. Hyams. "Experience with FNN Models for Medium Term Power Demand Predictions". IEEE Trans. on Power Systems, Vol. 14, No. 2, pp. 538-546, May 1999.
- [4] I. Derzga, S. Rahman. "Input Variable Selection for ANN-Based Short-Term Load Forecasting". IEEE Trans. on Power Systems, Vol. 13, No. 4, pp. 1238-1244, November 1998.
- [5] Diego Carmona, Miguel A. Jaramillo, Eva González, J. Antonio Álvarez. "Electric energy demand forecasting with neural networks". IECON-2002. 28th Conference of the IEEE Industrial Electronics Society. Sevilla (Spain). 5-8 November 2002
- [6] Hornik, K., Stinchcombe M. and White, H. "Multilayer Feedforward Networks are Universal Approximators". Neural Networks, Vol. 2, 1989, pp. 359-366.
- [7] C. M. Bishop. "Neural Networks for Pattern Recognition". *Oxford University Press*, 1995.
- [8] R. Reed. "Pruning Algorithms. A Survey". *IEEE Trans. on Neural Networks*, Vol. 4, No. 5, pp. 740-747, September 1993.